(12) SAM

TECHNICAL REPORT RD-SE-95-2

COMPUTATIONAL COMPLEXITY AND ENCRYPTION

John R. Coward
System Engineering and Production Directorate
Research, Development, and Engineering Center

FEBRUARY 1995

U. S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35898-5000

Approved for public release; Distribution is unlimited.

DESTRUCTION NOTICE

FOR CLASSIFIED DOCUMENTS, FOLLOW THE PROCEDURES IN DoD 5200.22-M, INDUSTRIAL SECURITY MANUAL, SECTION II-19 OR DoD 5200.1-R, INFORMATION SECURITY PROGRAM REGULATION, CHAPTER IX. FOR UNCLASSIFIED, LIMITED DOCUMENTS, DESTROY BY ANY METHOD THAT WILL PREVENT DISCLOSURE OF CONTENTS OR RECONSTRUCTION OF THE DOCUMENT.

DISCLAIMER

THE FINDINGS IN THIS REPORT ARE NOT TO BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION UNLESS SO DESIGNATED BY OTHER AUTHORIZED DOCUMENTS.

TRADE NAMES

USE OF TRADE NAMES OR MANUFACTURERS IN THIS REPORT DOES NOT CONSTITUTE AN OFFICIAL ENDORSEMENT OR APPROVAL OF THE USE OF SUCH COMMERCIAL HARDWARE OR SOFTWARE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
Distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
TR–RD–SE–95–2

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Sys. Eng. and Prod. Directorate
Res., Dev., and Eng. Center

6b. OFFICE SYMBOL *(If applicable)*
AMSMI–RD–SE–EA

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Commander, U. S. Army Missile Command
ATTN: AMSMI–RD–SE–EA
Redstone Arsenal, AL 35898

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Computational Complexity and Encryption

12. PERSONAL AUTHOR(S)
John R. Coward

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM ________ TO Jan 95

14. DATE OF REPORT *(Year, Month, Day)*
February 1995

15. PAGE COUNT
20

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Data Encryption Standard (DES)   Public Key Cryptography (PKC)
Polynomial Hierarchy   Algorithms

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Complexity theory has had a major impact on the design of encryption algorithms. Number theory has also made important contributions and is now the focus of much research in the cryptographic community. Work in theoretical computer science has lead to the development of a polynomial hierarchy used to classify the computational complexity of problems which are difficult to solve. NP–Complete problems have proven to be an excellent vehicle for the development of encryption algorithms. Asymmetrical (Public Key) encryption uses complexity theory for the design of algorithms that provide information security and integrity. Encryption algorithms based on complexity theory are relatively new, and questions remain about the security and integrity of these algorithms. Complexity theory has proven to this point to be a powerful ally in the design of encryption algorithms, but only time will tell if the polynomial hierarchy is real, or if advances in mathematics and computer science will render these algorithms insecure.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
John R. Coward

22b. TELEPHONE *(Include Area Code)*
(205) 842–9258

22c. OFFICE SYMBOL
AMSMI–RD–SE–EA

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. ENCRYPTION ALGORITHMS

Encryption algorithms have been widely used in the Military and Diplomatic communities to transform information, often called clear text, into unintelligible information called cipher text. These algorithms are most often mathematical transformations that use a secret key to encrypt information. The design of modern encryption algorithms are usually developed in one of two ways (Fig. 1). The first method looks at the design of encryption schemes from the code breakers point of view, and seeks to use techniques that place a premium on resources that are in short supply to the code breakers. The Data Encryption Standard (DES) is based on this premise. Its use of substitution/permutation schemes, sometimes referred to as product ciphers, tax the computing power, time, memory, and money of aspiring code breakers. The term *Work Factor* is often used to assess the strength of encryption algorithms and determine the amount of resources required to break an algorithm. If the amount of money needed to break a code is more than the value of the information, then the algorithm is sufficiently strong. Breaking DES with its 56 bit key would be a formidable challenge (Fig. 2). Using a brute force approach, by trying every possible key, and testing one million keys per second would take more than 2000 years. Statistically, there would be a 50 percent chance of breaking the code in approximately 1000 years. The other technique is to base the encryption algorithm on solving a problem that is considered intractable, and has throughout the history of mathematics and computer science proven difficult, if not impossible to solve.

DESIGN PHILOSOPHY	EXAMPLE
BASED ON CODEBREAKER RESOURCES	DES
BASED ON COMPUTATIONAL COMPLEXITY	RSA

Figure 1. Design of Encryption Algorithms

$$2^{56} = \left(\frac{7.2 \times 10^{16}\ \text{KEYS}}{1 \times 10^{6}\ \text{KEYS/SEC}}\right)\left(\frac{1\ \text{min}}{60\ \text{SEC}}\right)\left(\frac{1\ \text{HR}}{60\ \text{MIN}}\right)\left(\frac{1\ \text{DAY}}{24\ \text{HRS}}\right)\left(\frac{1\ \text{YR}}{365\ \text{DAYS}}\right)$$

$$= 2284.9\ \text{YRS}.$$

Figure 2. Brute Force Attack on DES

II. NUMBER THEORY

Number theory which was once considered an esoteric field of mathematics is making important contributions to the development of modern encryption systems. Number theory is primarily focused on the juggling and manipulation of whole numbers. Number theory has long been considered to be the purest form of mathematics, but until recently it has had few practical applications. The last few decades, with the widespread use of computers, number theory has grown in prominence and proven to have many practical applications. The generation of random numbers, prime numbers and complexity theory are all part of number theory domain and have many applications to the designers of crypto systems. The generation of true random numbers plays a critical role in the design of secure encryption keys. Prime numbers are very important because of the difficulty involved in the factoring of large prime numbers provides the security of many Public Key Encryption (PKE) algorithms. Complexity theory provides the mathematical foundation for the design and testing of encryption schemes. Number theory has evolved from an esoteric branch of mathematics to become the focus of intense research by the designers of encryption algorithms.

III. POLYNOMIAL HIERARCHY

Researchers working in the field of theoretical computer science have developed a hierarchy to classify the computational complexity of problems that are difficult to solve. This ranking is often called the polynomial hierarchy (Fig. 3). The bottom of the hierarchy consists of problems which are the easiest to solve. The number of computations required to solve these problems are a polynomial function of the size of the problem. An example of this is n**3. The number of steps needed to solve this problem grows relatively slowly as n increases. This class of problems are commonly called P (Polynomial Time) problems because the number of steps needed to solve these problems are proportional to the amount of computational time needed to solve these problems. The next step in the polynomial hierarchy are the Nondeterministic Polynomial (NP) Problems. These problems are inherently more difficult, since given a potential solution to the problem, it can be checked in polynomial time, but no known polynomial time algorithm exists to solve these problems. These NP Problems are often referred to as exponential time algorithms since the number of steps needed to solve these problems increases as an exponential as the n, the size of the problem increases. The top of the hierarchy is composed of NP–Complete problems which are considered the most difficult of the NP class, and are the basis of encryption algorithms that capitalize on computational complexity.

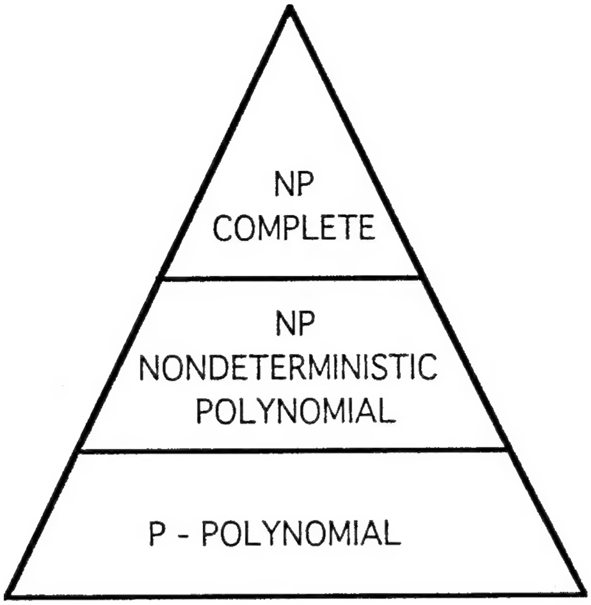

Figure 3. Polynomial Hierarchy

IV. COMPUTATIONAL COMPLEXITY

Encryption algorithms based on NP–Complete problems are making important contributions to the development of complexity theory. NP–Complete problems are a well known class of problems and are the focus of much research in theoretical computer science. Theoretical computer science seeks to determine what types of problems can be solved by computer and which classes of problems can not be solved using computers. Throughout history, scientists and mathematicians have struggled with problems which seemed insolvable. Many of these problems are now being solved with the help of high speed computing. There still are many problems which can not be solved, and may never be solved no matter how many advances are made in high speed computing. These problems are often called intractable or NP problems and are the focus of much research in the field of theoretical computer science and by designers of encryption systems. The discovery of a polynomial time algorithm which would solve these NP–Complete problems is considered akin to finding the Holy Grail of computer science. This would constitute a major breakthrough, and would have a major impact on both theoretical computer science and the development of complexity based encryption algorithms. The complexity of these NP–Complete problems increases dramatically as the size of the problem increases. To put it another way, as the problem grows sufficiently large, an NP–Complete running on the fastest super computer would not be solved before a polynomial time algorithm running on the slowest personal computer (Figs. 5 and 6). By capitalizing on problems which are inherently difficult, encryption is providing real world application of computationally complex problems. Complexity is generally considered the bane of engineers and scientist trying to solve complicated technical problems, but have proven to be an excellent vehicle for the development of encryption algorithms. Cryptography is playing an important role in complexity theory and gaining respectability, since it provides a practical application of complexity theory, and the theory of a polynomial hierarchy. The use of these computationally complex problems as the foundation of encryption schemes can result in unbreakable encryption codes. Using problems that are computationally complex as the foundation for encryption schemes is more difficult than it seems. Special care must be taken when choosing a computationally complex problem as the basis for an encryption scheme. Complexity in itself is not a panacea, because these problems are difficult, if not impossible to prove secure. An encryption algorithm can not be judged on average–case or worst–case complexity, but on the complexity of the easiest solution. This is because that one instance where the problem can be solved by polynomial time algorithm could render the encryption algorithm insecure. The knapsack problem was initially thought to be secure against cryptographic attack, but was later proven insecure. The focus of computationally complex algorithms are on the lower bounds of the problem (Fig. 6). The security of a complexity based algorithm would require that the lower bound, or easiest solution of the problem could not be solved in polynomial time.

Number of Inputs Time Complexity Function	10	20	30	40	50
n^3	.001 sec	.008 sec	.027 sec	.064 sec	.125 sec
3^n	.059 sec	58 min	6.5 yrs	4 x 10^5 yrs	2 x 10^{10} yrs

Figure 4. Complexity Time Functions

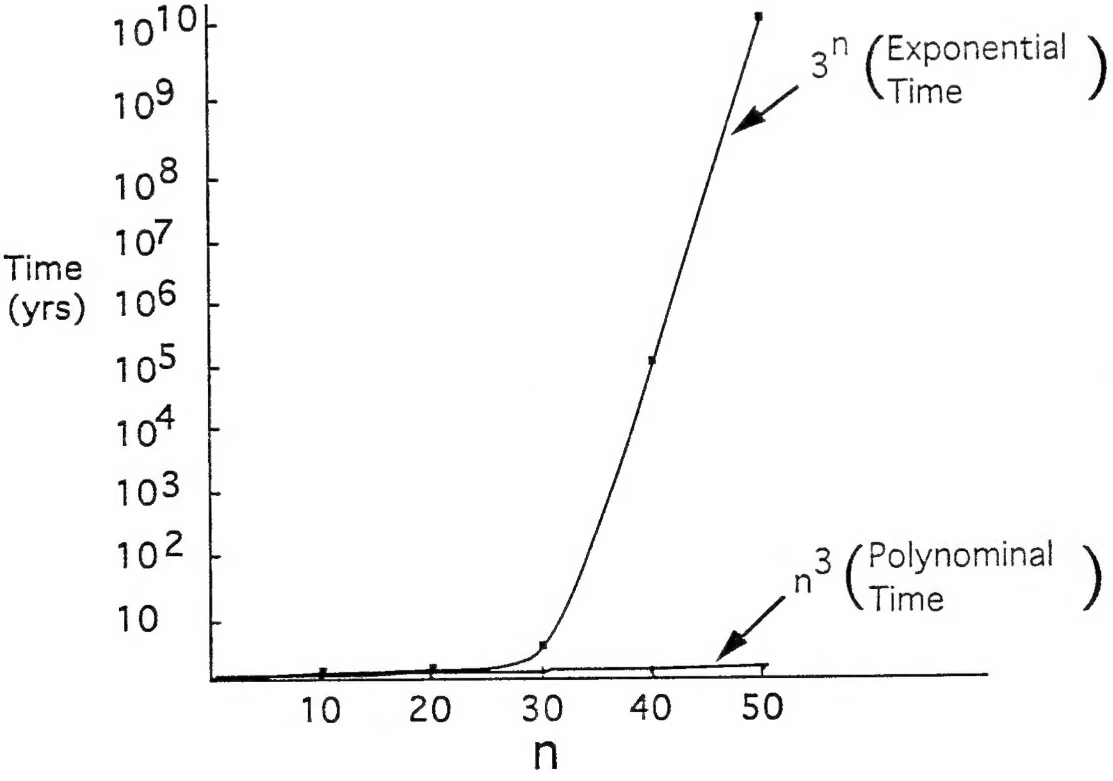

Figure 5. Polynomial Versus Exponential Time

PROBLEMS	UPPER BOUND	LOWER BOUND
P PROBLEMS	N^K	________
NP PROBLEMS	________	K^N

K = CONSTANT

Figure 6. Upper/Lower Bounds of P and NP Problems

V. ASYMMETRICAL (PUBLIC KEY) ENCRYPTION

The formal link between complexity theory and encryption was complete in 1976, when Whit Diffie and Martin Hellmann proposed in their landmark paper, "New Directions in Cryptography", to use NP Complete problems as the basis for asymmetrical encryption algorithms. These asymmetrical encryption algorithms are often referred to as Public Key Cryptography (PKC), which uses a pair of keys, one public and one private (Fig. 7). Public keys do not need to be kept secret and may be published in a book for use by other users. This concept relies on the fact that even though a potential code breaker knows the public key, it is infeasible to deduce the private key. This is because the two keys are related by an NP–Complete problem and attempting to determine one key from another is tantamount to solving this computationally complex problem. These algorithms contain a trapdoor one way function which makes the problem easy to solve oneway, but computationally infeasible to reverse. For example, in the problem below, it is easy to compute Y given X, but much more computationally intensive to compute X given Y.

$$Y = X^5 + 7X^4 + 25X^3 + 78X^2 + 413$$

PKC allows not only the transfer of secret information but can be used to ensure it's integrity as well. If person B used A's public key to encrypt a message, then only person A can decrypt the message (security). If person A encrypts a message with his private key then anyone with access to A's public key can decrypt the message and be sure that it was A that sent the message (authentication). The most widely used PKC algorithm is the Rivest, Shamir, Adleman (RSA) algorithm. The RSA algorithm's security lies in the complexity of factoring very large prime numbers, a problem well known by number theorists, and long considered intractable. The message is first converted into a numeric representation, which is a very large number. It is then raised to a high power and then reduced to a modulo of another number. The modulus is the product of the two large prime numbers, and the Holy Grail for any potential code breaker, since in order to crack the code, the adversary needs to find the two large prime numbers that made up the modulus. The security of the code essentially rests with the generation of a large modulus from two large primes, since the amount of computer time needed to solve the problem would render it intractable and unable to be solved in polynomial time.

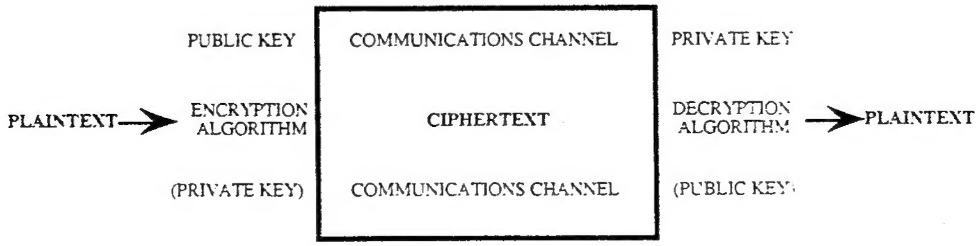

Figure 7. Asymmetrical Encryption

VI. CONCLUSIONS

This paper attempted to explore the development of encryption algorithms based on computational complexity. The use of complexity based encryption has to date shown great promise as seen by the development of PKC. Encryption algorithms based on complexity theory are relatively new and a nagging question exists in both encryption and theoretical computer science, as to whether the polynomial hierarchy is real or if all classes are the same and can be solved in polynomial time. There is a great deal of research ongoing to determine if P = NP, and this has become a vexing problem in both the encryption and theoretical computer science communities. Advances in mathematics may eventually solve this debate, but proving this conjecture true would have major repercussions in the design of modern encryption algorithms.

REFERENCES

1. Beker, H. and Piper F., Cipher Systems, John Wiley and Sons Inc., New York, 1982.

2. Boyd, C., "Modern Data Encryption", Electronics and Communications Engineering Journal, October 1993.

3. Cambell, Carl M., "Design and Specification of Cryptographic Capabilities" IEEE Communications Society Magazine, November 1978, Vol. 16, No 6, pp. 15–19.

4. Coward, John R., "Information Security for Unmanned Systems", Unmanned Systems, Spring 94, No. 2, pp. 12–15.

5. Diffie, Whitfield and Hellman, Martin E., "New Directions in Cryptography", IEEE Transactions on Information Theory, Vol. IT–22, No. 6, November 1976.

6. Dror, Asael, "Secret Codes", BYTE Magazine, June 1989.

7. Guterl, Fred, "Suddenly, Number Theory Makes Sense to Industry", Business Week, 20 June 94, pp. 172–174.

8. Kolata, G., "Must Hard Problems Be Hard", Science, 1985, v228 pp. 479–481.

9. National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standard, Publication No. 46, Jan. 1977.

10. Riek, Justas, "Applications of Public–Key Cryptography to Computer/Communications Security", Data Systems Division, Grumman Corporation, 1987.

11. Traub, Joseph F. and Wozniakowski, Henryk, "Breaking Intractability", Scientific American, January 1994.

BIOGRAPHY

John R. Coward works as an electronics engineer at the U. S. Army Missile Command in Huntsville, Alabama. He has worked on both Air Defense missiles and Unmanned systems. His primary interests include computer and information security, encryption, worms/viruses and TEMPEST.

ACRONYMS

DES	Data Encryption Standard
NP	Nondeterministic Polynomial
P	Polynomial
PKC	Public Key Encryption
RSA	Rivest, Shamir, Adleman

INITIAL DISTRIBUTION LIST

		Copies
IIT Research Institute ATTN: GACIAC 10 W. 35th Street Chicago, IL 60616		1
AMSMI–RD		1
AMSMI–RD–AC–AD,	D. Peterson	1
	S. Young	1
	L. Levitt	1
AMSMI–RD–AS,	B. Pittman	1
AMSMI–RD–AS–MM		1
AMSMI–RD–AS–PM		1
AMSMI–RD–BA–C3I		1
AMSMI–RD–BA–TU		1
AMSMI–RD–CS–R		15
AMSMI–RD–CS–T		1
AMSMI–RD–GC–S		1
AMSMI–RD–SE–EA,	J. Coward	5
AMSMI–RD–SE–ES		1
AMSMI–RD–SE–PE,	A. Roberts	1
	S. Stephens	1
AMSMI–RD–SI,	D. Trenkle	1
AMSMI–RD–SS–SP,	K. Flynn	1
AMSMI–RD–SS–ST,	B. Walker	1
AMSMI–RD–WS–LS,	K. Jordan	1
AMSMI–GC–IP,	Mr. Fred Bush	1